Performance Analysis of the ARL Linux Networx Cluster

George Petit and Steven R. Thompson
US Army Research Laboratory (ARL)/Raytheon Company, Aberdeen Proving Ground, MD
{gpetit, thompson}@arl.army.mil

Abstract

Within the past year, a 256-processor I686 Linux Cluster was installed at the Army Research Laboratory (ARL) Major Shared Resource Center (MSRC) to augment the center's current unclassified scientific application processing capabilities. The purpose of this paper is to provide a comparative analysis of wall-clock-time performance of this system and the other unclassified HPC platforms currently installed at the ARL MSRC. A suite of vendor applications currently receiving significant utilization on the ARL platforms will be used to perform this analysis. The other existing unclassified HPC platforms at ARL include: an SGI 3800 with 512 processors, an IBM SP3 with 1024 processors, and an IBM SP4 with 128 processors. The following application codes will be used: CTH, CFD++, OVERFLOW, GAMESS, COBALT, LS_DYNA and FLUENT. Each code will be run using 16, 32, 48, 64, 80, 96, 112, and 128 processors. Using these timing metrics we will analyze the appropriateness of the I686 architecture for use as a large-scale distributed computing platform for each of these scientific application codes.

1. Introduction

The purpose of this study was to evaluate the performance of the ARL Linux Networx cluster compared to the more traditional high-performance computing platforms currently being utilized for DoD engineering applications. Using these results we will demonstrate the suitability to the emerging Linux network architectures as viable HPC platforms in the scientific and engineering computational environment. This will be accomplished by executing a suite of benchmarks of representative ARL codes, and comparing run-time performance of each code on the ARL architectures.

The Army Research Laboratory MSRC's 256-processor I686 Linux Cluster, 512-processor SGI Origin 3800, 1024-processor IBM SP3, and 128-processor IBM SP4 were used to perform benchmark runs on all codes in the suite. CTH, COBALT, and GAMESS were run with configurations of 16, 32, 48, 64, 80, 96, 112, and 128 processors. OVERFLOW was run with configurations of 8, 16, 32, and 64 processors. CFD++ was run with configurations of 8, 16, 32, 64, and 128 processors. FLUENT was run using 64 processors. LS-DYNA had to be excluded from the benchmark suite because the vendor was unable to provide a binary that would run properly in parallel on the ARL I686 Linux cluster. All benchmarks on the I686 Linux Cluster were executed using 2 processors per node. All benchmarks on the IBM SP3 were executed on dedicated 16-processor nodes. All 16, 32, 64, 96, and 128 processor benchmarks on the IBM SP4 were executed using dedicated 32-processor nodes. 48, 80, and 112 processor benchmarks used processors selected by SGE. All benchmarks on the Origin 3800 were executed using IRIX cpusets. The overall wall-clock time for each processor configuration was recorded for each benchmark, except for CFD++ which uses time-steps/hour, and comparative results are provided within this paper.

2. Benchmark Code Suite

CTH: This code is used for modeling multidimensional, multimaterial, large deformation, strong shock wave propagation problems in solid mechanics. It uses advanced numerical methods coupled with advanced material models to model the non-linear behavior of materials subjected to large deformations under high strain rate loading. The Eulerian finite volume (FV) code employs a two-step solution scheme: a Lagrangian step and a remap step. The conservation equations are replaced by explicit FV equations that are solved in the Lagrangian step. The remap step replaces multidimensional equations with a set of one-dimensional equations. The remap, or advection, step is based on a second order accurate Van Leer scheme[1]. The input used for this benchmark is the standard input provided in the HPCMP TI-04 benchmark suite.

0-7695-2259-9/04 $20.00 © 2004 IEEE

CFD++: CFD++ is a general-purpose Computational Fluid Dynamics code for accurate and efficient flow simulations. Its unified-grid, unified-physics and unified-computing methodology applies to all flow regimes, all types of mesh and cell topologies. The input used for these benchmarks defines a 4 million grid point mesh [2].

COBALT: This code, widely used in a variety of both internal and external flow simulations, is a finite volume unstructured-grid Eulerian/Navier-Stokes solver. COBALT uses a finite-volume, cell-centered, first-order accurate in space and time, exact Riemann solver [1]. The input used for these benchmarks was the standard input provided in the HPCMO TI-04 benchmark suite.

FLUENT: FLUENT is a widely-used CFD code for simulation, visualization and analysis of fluid flow, heat and mass transfer, and chemical reactions. The input used for these benchmarks defines a missile with grid fins consisting of seventeen million cells [3].

GAMESS: GAMESS (General Atomic and Molecular Electronic Structure System) is a program for *ab initio* quantum chemistry. It can compute wave functions ranging from RHF, ROHF, UHF, GVB and MCSCF, with CI and MP2 energy corrections for some of these. Analytic gradients are available for these self-contained field functions, for automatic geometry optimization, transition state searches, and reaction path following [1]. The input used for these benchmarks was the standard input provided in the HPCMO TI-04 benchmark suite.

OVERFLOW: This code is based on Overset structured grids (Chimera). Geometry complexity is reduced to a set of relatively simple overlapping body-fitted grids and topologically simple background grids. The structure of the individual grid components facilitates viscous boundary layer resolution, implicit time-integration algorithms, and efficient use of computer memory[1]. The input used for these benchmarks is the Trapwing-7m data files provided in the HPCMP TI-02 benchmark suite.

3. Hardware/Software Configuration

Table 1 provides the hardware configuration for the ARL Linux Networx cluster, SGI Origin 3800, IBM SP3, and IBM SP4 used for this study.

Table 1. System hardware configurations

System	Processor Type	Processor Speed	No. of Processors	Processors per node	Memory per Node	Communication Speed	Storage Space
Linux NetworX Cluster	Intel IA-32	3.06 GHz	256	2	2 GB	2 GB/sec	10 TB
SGI Origin 3800	R12000	400 MHz	512	4	3 GB	1600 MB/sec	1 TB
IBM SP3	Power 3	375 MHz	1024	16	16 GB	500 MB/sec	3 TB
IBM SP4	Power 4	1.7 GHz	128	32	32 GB	2GB/sec	6 TB

The following operating system and application support software were available for building and running the application codes on the ARL Linux Networx Xeon cluster: GNU C, C++, and FORTRAN 77/90 compilers for both Ethernet and Myrinet applications, PGI C, C++, and FORTRAN 77/90 compilers for both Ethernet and Myrinet applications, INTEL C, C++, and FORTRAN 77/90 compilers for both Ethernet and Myrinet applications, the MPICH message-passing software library. All applications on the Linux Networx cluster were built for use on the Myrinet communication hardware. On the Origin 3800, IBM SP3, and IBM SP4 the standard Unix-based operating system and application support software were used. Execution of the benchmarks on all systems was performed using the SGE batch scheduler.

4. Application Performance Results and Analysis

Tables 2 through 7 lists the run-time results obtained for each of the benchmark for each platform. Figures 1 through 6 provide a graphical comparison of the run-time results of the benchmarks on the ARL Linux Networx cluster, SGI Origin 3800, IBM SP3, and IBM SP4.

4.1. CTH.

CTH was run successfully on all platforms. The run times indicate the IBM SP4 clearly outperformed all other architectures. However, the Linux Cluster significantly outperformed the IBM SP3 and SGI Origin 3800. The SGI Origin 3800, representing the oldest of the architectures, not unexpectedly lagged behind the other platforms in performance. It is also noteworthy that parallel performance peaked at 96 processors for the IBM SP4, while performance improvement continued through all 128 processors for the other platforms. This is indicative that the CPU performance on the IBM SP4 outstripped the ability of the communication switch to transfer data between nodes using the higher number of processors.

Table 2. CTH performance statistics (wall-clock seconds)

No. of Processors	Linux Networx Cluster	IBM SP4	IBM SP3	SGI Origin 3800
16	6400	2748	9917	11937
32	2914	1541	4827	5626
48	2180	1260	3591	4872
64	1838	1055	3036	3944
80	1803	1214	2976	3036
96	1529	1090	2775	2472
112	1407	1124	2720	3401
128	1313	1086	2211	2539

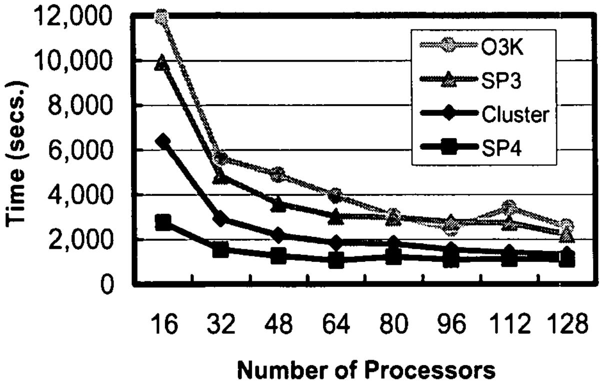

Figure 1. CTH run times

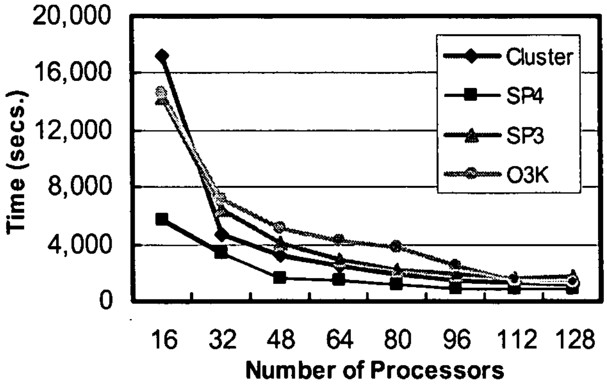

Figure 2. COBALT run times

4.2. COBALT.

COBALT was run successfully on all platforms. The Linux Cluster again outperformed the IBM SP3 and Origin 3800 except at 16 processors. The IBM SP4 once again outperformed all other architectures while also exhibiting a fall-off in performance at the higher processor count. This performance fall off is attributed to the memory footprint of the code at this number of processors. The memory requirement of each of the two processes on the node exceeded half the memory available. This caused significant use of swap space which seriously degraded performance on the nodes affected. This result underscores the importance of not exceeding the available memory space on Linux Cluster nodes.

Table 3. COBALT performance statistics (wall-clock seconds)

No. of Processors	Linux Networx Cluster	IBM SP4	IBM SP3	SGI Origin 3800
16	17233	5691	14364	14528
32	4663	3348	6480	7221
48	3171	1653	4036	5071
64	2480	1493	2906	4212
80	1889	1149	2214	3781
96	1515	904	1861	2422
112	1282	804	2720	1483
128	1163	816	2211	1268

4.3. GAMESS.

GAMESS was successfully run on the Linux cluster, the IBM SP3, and the IBM SP4. However, on the Origin 3800 the jobs failed with an error indicating that eigenvalue calculations were not converging. This anomaly precluded the use of this platform for this benchmark. The results using the remaining three platforms follow closely the previous benchmarks analyzed. Note, however, that the IBM SP4 outperforms the Linux Cluster by better than two-to-one at 16 processors, but less than two-to-one using 32 or more processors. In fact, the Linux Cluster actually outperformed the IBM SP4 at 128 processors. The better performance of the IBM SP4 using 16 processors can be attributed to the fact that since all the processes are on one dedicated node there is no inter-nodal communication required to share data between processes, thus significantly reducing communication overhead. At 32 processes, although all the processes are on the same node, there is much more process swapping to handle system request since all the processors are being utilized by the benchmark code.

Table 4. GAMESS performance statistics (wall-clock seconds)

No. of Processors	Linux Networx Cluster	IBM SP4	IBM SP3
16	5278	2477	9940
32	2670	1570	6523
48	1970	1219	5421
64	1730	1120	4785
80	1394	1040	4465
96	1174	1029	4303
112	1152	955	4123
128	1144	1221	4051

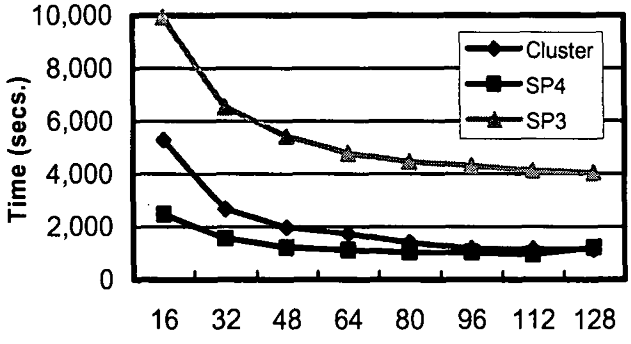

Figure 3. GAMESS run times

4.4. FLUENT.

FLUENT was successfully run on all platforms. However, the code exhibited very little scalability. Hence, for the purpose of this study a comparison of performance using 64 processors was analyzed. The results conform to previous performance measurements except the Origin 3800, which exhibited extremely poor performance compared to the other platforms benchmarked.

Table 5. FLUENT performance statistics (wall-clock seconds)

No. of Processors	Linux Networx Cluster	IBM SP4	IBM SP3	SGI Origin 3800
16	NA	NA	NA	NA
32	NA	NA	NA	NA
48	NA	NA	NA	NA
64	5500	4913	6352	52425
80	NA	NA	NA	NA
96	NA	NA	NA	NA
112	NA	NA	NA	NA
128	NA	NA	NA	NA

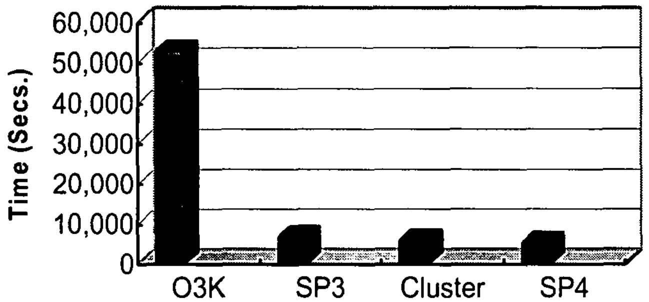

Figure 4. FLUENT run times

4.5. OVERFLOW.

OVERFLOW was successfully run on the Linux cluster, IBM SP4 and IBM SP3. However, the jobs failed on the Origin 3800 with a message indicating a bus error, precluding its use for this benchmark. Also, although the capability is provided with the code to create the required MPIINP namelists for 48, 80, 96, 112, and 128 processors, the code would not successfully run using MPIINP namelists generated for these numbers of processors. Hence, the benchmarks were run using the MPIINP namelists for 8, 16, 32, and 64 processors provided with the other input files. In general, the results follow the trend of previous benchmark analysis. However, the performance of the Linux Cluster is only slightly better than the IBM SP3, whereas in previous benchmarks it was close to or exceeded twice the performance of the IBM SP3.

Table 6. OVERFLOW performance statistics (wall-clock seconds)

No. of Processors	Linux Networx Cluster	IBM SP4	IBM SP3
8	2173	764	2435
16	1289	500	1609
32	576	231	576
64	375	150	619

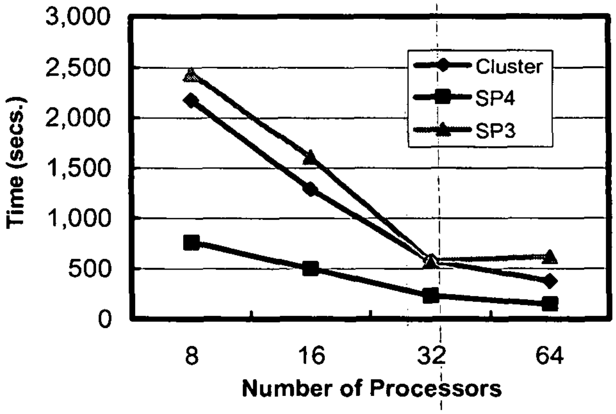

Figure 5. OVERFLOW run times

4.6. CFD++.

CFD++ was run successfully on all platforms. Note that for this benchmark we are using results provided by J. Sahu. The measurement parameter utilized (Time Step/Hour vs. Overall Run Time) is slightly different, however the results provide the same qualitative comparison. For this benchmark the Linux Cluster slightly outperforms the IBM SP4. Also, there is a more dichotomous difference in performance between the top two performers (the Linux Cluster and IBM SP4) and the bottom two performers (the IBM SP3 and SGI Origin 3800).

Table 7. CFD++ performance statistics (time steps/hour)

No. of Processors	Linux Networx Cluster	IBM SP4	IBM SP3	SGI Origin 3800
8	60	93	40	30
16	255	190	78	55
32	500	409	150	120
64	843	735	300	257
128	1162	1094	400	357

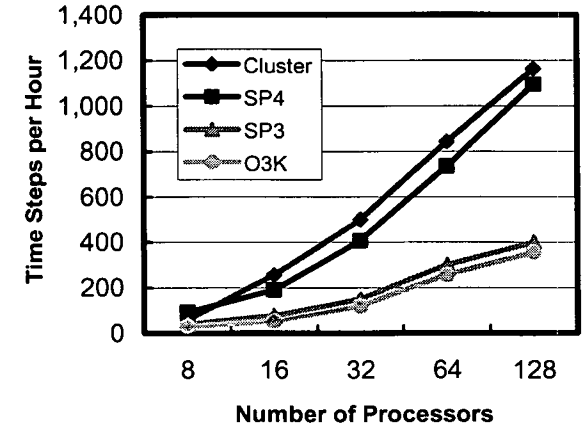

Figure 6. CFD++ run times

5. Conclusions

Based on the above analysis it is clear that the ARL Linux Networx Xeon Cluster is a capable and important addition to the ARL MSRC HPC suite. It has proven itself capable of significantly outperforming the IBM SP3 and SGI Origin 3800 on a variety important application codes representing several computation technology areas important to DoD research. It also performed well against the IBM SP4, approaching the IBM SP4's performance in several application codes at the larger number processors, and outperforming the IBM SP4 on the CFD++ benchmark. Its current main shortcomings are its 32-bit architecture and limited memory space per node board. ARL plans to address these shortcomings through future upgrades and acquisitions.

Acknowledgements

We would like to express our appreciation to Phil Mathews of the ARL MSRC System Administration staff and Tom Kendall of ARL for their efforts in expediting many of these benchmark jobs through the batch queue system.

This work was supported in part by a grant of computer time from the DoD High Performance Computing Modernization Program at the ARL MSRC.

References

1. HPC Modernization Program Software Listings, www.hpcmo.hpc.mil/CHSSI/software.html.

2. CFD++ input and results provide by Jubaraj Sahu of US Army Research Laboratory, Aberdeen Proving Ground, MD.

3. FLUENT input provided by Dr. James Despirito of US Army Research Laboratory, Aberdeen Proving Ground, MD.